

A 1200 BAUD KC STD. INTERFACE FOR THE SWTP 6800

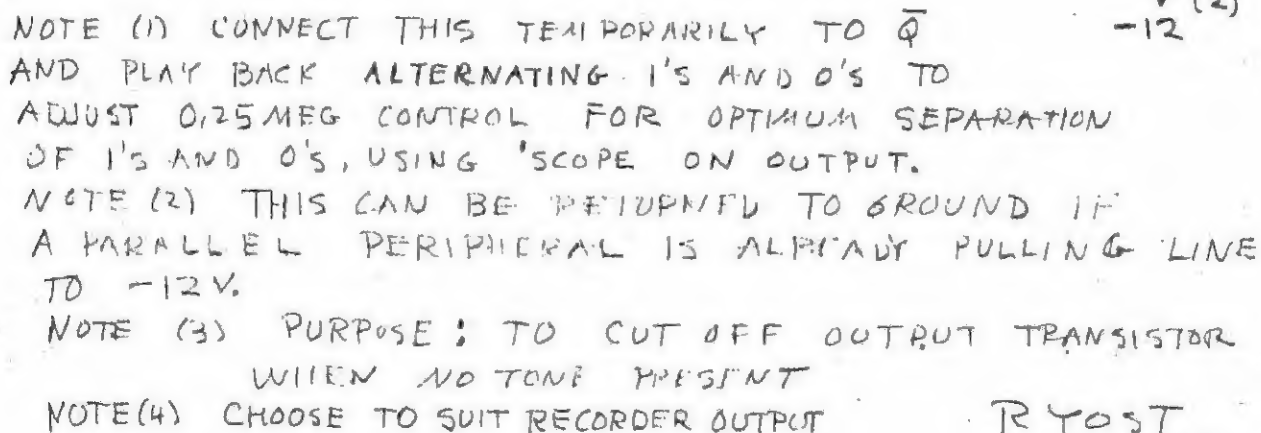
EVER SINCE I GOT MY SYSTEM UP AND RUNNING, A 1200 BAUD CASSETTE MODEM HAS BEEN LIKE MT. EVEREST FOR ME - - I WANTED (TO CLIMB) IT - BECAUSE IT WAS THERE! OF COURSE, IT DID'NT BOTHER ME TOO MUCH AT FIRST. AT THAT POINT, ANY SORT OF CASSETTE INTERFACE WAS GOOD ENOUGH! SO I THREW TOGETHER A CRUDE 300 BAUD "MODEM" USING THE POPULAR KANSAS CITY STANDARD FREQUENCIES, 2400 HZ FOR A 1 AND 1200 HZ FOR A 0.

THE CIRCUIT, SHOWN IN FIGURE 1, USES A QUAD NOR GATE AND A PAIR OF DUAL TYPE D FLIPFLOPS. THE 4800 CLOCK SIGNAL FROM THE COMPUTER IS FIRST DIVIDED BY 2 TO OBTAIN 2400 HZ FOR THE 1'S. DURING LOGIC 0'S (RS232 LINE HIGH) THE RESET OF FF-2 IS HELD LOW, SO IT FURTHER DIVIDES THE FREQUENCY BY 2 AGAIN TO OBTAIN 1200 HZ. DURING LOGIC 1'S (RS232 LINE LOW) THE NOR GATE RESETS FF-2 ON THE NEGATIVE TRANSITIONS OF ITS CLOCK INPUT, SO FF-2'S OUTPUT IS 2400 HZ. THE SQUARE WAVES CAN BE FED DIRECTLY TO THE TAPE RECORDER THROUGH A LEVEL ADJUSTING POTENTIOMETER.

ON PLAYBACK, EXCESSIVE DIFFERENTIATION AND DISTORTION OF THE SQUAREWAVES IS AVOIDED BY LOADING THE TAPE RECORDER OUTPUT WITH A RESISTOR 2 TO 3 X'S THE NORMAL LOAD. AN 8 OHM SPEAKER CAN BE REPLACED BY ABOUT 15 TO 22 OHMS, WHILE A 30 OHM "EAR" OUTPUT COULD BE LOADED BY 68 TO 100 OHMS. THIS RESISTIVE LOAD SHUNTS THE INDUCTANCE OF THE OUTPUT TRANSFORMER OF THE TAPE RECORDER, PREVENTING ITS RISING IMPEDANCE WITH FREQUENCY FROM DISTORTING THE SQUARE WAVES. Q1 SQUARES UP THE TAPE RECORDER OUTPUT. IT IS DC BIASED SO THAT IN THE ABSENCE OF SIGNAL ITS COLLECTOR GOES LOW, WHICH SETS FF-3, CUTTING OFF Q-2 AND "RELEASING" THE RS232 LINE. WITH TAPE SIGNAL PRESENT, THE DETECTOR CIRCUIT ATTACHED TO THE SET TERMINAL OF FF-3 REMOVES THE SET INPUT. FOR 2400 HZ, THE RESET INPUT REMAINS BELOW THE TRIGGER LEVEL AND THE CLOCK INPUT PULLS Q OF FF-3 HIGH, CUTTING OFF Q-2, AND ALLOWING THE LINE TO GO LOW. WHEN THE 1200 HZ SIGNAL IS PRESENT, THE RESET EXCEEDS THE TRIGGER LEVEL BEFORE THE CLOCK EDGE AND INHIBITS IT - THUS Q REMAINS LOW, WHICH TURNS ON Q-2 AND PULLS THE LINE HIGH.

THIS SIMPLE CIRCUIT SUFFICED FOR QUITE A WHILE, AND JOE DUBNER, OF ACS, AND I WERE ABLE TO EXCHANGE TAPES, IN SPITE OF SLIGHT DIFFERENCES IN OUR CLOCK FREQUENCIES AND TAPE RECORDER SPEEDS.

THEN I BEGAN TO CONSIDER MAKING TAPES FOR SALE TO OTHERS, AND REALIZED THAT THEY WERE GOING TO HAVE TO BE SUITABLE FOR PLAYBACK ON A VARIETY OF RECORDERS, SOME PROBABLY NOT OF THE QUALITY OF MINE. I USED DON LANCASTER'S CLASSIC ARTICLE IN THE MARCH '76 BYTE MAGAZINE AS MY "BIBLE", AND DECIDED THAT SINEWAVE RECORDING WAS THE ONLY RELIABLE TYPE. I BUILT A VARIATION OF DON'S "BIT BOFFER" AND STARTED USING IT. HOWEVER, AFTER TRY-



RYOST
265-8047

ING A CIRCUIT LIKE HIS FOR RECOVERING THE CLOCK, I BECAME DISEN-
 CHANTED WITH ITS TRADEOFF BETWEEN ADEQUATELY WIDE CLOCK PULSES AND SPEED
 TOLERANCE, AND WITH ITS SENSITIVITY TO RANDOM ERRORS.
 I WANTED TO USE DON'S SINEWAVE SYNTHESIZER, BUT DIDN'T
 WANT TO INSTALL SWITCHES IN MY COMPUTER TO SWITCH TO THE
 19.2 KHZ CLOCK FOR IT, SO I DECIDED TO INCLUDE A PHASE
 LOCKED LOOP CLOCK GENERATOR RUNNING AT 19.2 KHZ, WITH TWO
 DIVIDERS TO OBTAIN 4800 HZ TO PHASE COMPARE WITH
 A 4800 HZ SIGNAL DERIVED FROM THE TAPE RECORDER'S ZERO CROSS-
 INGS.

IT WAS THEN THAT SOME OF THE PECULIARITIES OF THE DATA
 STREAM GENERATED BY MIKBUG'S SOFTWARE UART CAME TO LIGHT.
 BECAUSE A DATA WORD (A 0 START PULSE, 8 DATA BITS, AND A 1
 STOP PULSE) COULD START ON ANY ONE OF THE 4800 HZ CLOCK PULSES
 , THE 2400 AND 1200 HZ SIGNALS DERIVED FROM THE TAPE WERE NOT
 SYNCHRONOUS WITH THE DATA, THUS AS FREQUENCIES, THEY WEREN'T
 VERY STABLE. LANCASTER PROVIDED A SYNCH CONNECTION FROM HIS
 SINE SYNTHESIZER TO THE DIVIDER FF'S, WHICH GENERATED EXTRA
 CLOCK PULSES; THIS MADE A CLOCK RECOVERED FROM THE RECORDED
 DATA EVEN FURTHER FROM A STABLE 4800 HZ CLOCK SUITABLE FOR
 THE MIKBUG SOFTWARE UART.

NONETHELESS, AFTER CLEANING UP THIS CIRCUIT, I WAS ABLE
 TO GET IT WORKING WELL AT 300 AND 600 BAUD - BUT NEVER AT
 1200 BAUD! EVEN AT 600, I FOUND THAT IT WAS BEST NOT TO USE
 LANCASTER'S SYNCH CONNECTION.

RECENTLY, I DECIDED TO TRY AGAIN TO SCALE THE 1200 BAUD
 "PEAK". I DECIDED TO RELOCK THE DATA COMING OUT OF MIKBUG,
 USING FOR A CLOCK ONE OF THE SQUARE WAVE OUTPUTS OF THE SINE-
 WAVE SYNTHESIZER. THUS, REGARDLESS OF WHAT CYCLE OF THE 4800
 HZ CLOCK A WORD STARTED ON, RELOCKING
 IT WITH A D FLIPFLOP WOULD DELAY IT ENOUGH TO KEEP THE BIT EDGES
 SYNCHRONOUS WITH THE SINEWAVES. IN OTHER WORDS, I COULD FORCE
 THE DATA TO ALWAYS CHANGE AT, SAY, THE SINEWAVE ZERO CROSSINGS, OR
 AT OTHER CONTROLLED PHASES.

TO MY SURPRISE, THIS TOO FAILED TO WORK AT 1200 BAUD. I
 THOUGHT THE TROUBLE WAS IN MY DATA RECOVERY CIRCUITS, BUT WHEN
 DUBNER BROUGHT OVER A TAPE HE'D RECORDED AT 1200 BAUD AND I
 WAS ABLE TO SUCCESSFULLY LOAD IT INTO THE SWTP 6800 USING THE
 MIKBUG LOAD COMMAND, I BEGAN TO LOOK AT MIKBUG'S OUTPUT MORE
 CLOSELY.

DUBNER HAD USED AN ACIA RATHER THAN THE MIKBUG
 SOFTWARE UART. THAT CLUE LED TO DISCOVERY OF THE PROBLEM.
 AT 1200 BAUD, MIKBUG'S START PULSES WERE SOMETIMES 17 CLOCK
 PERIODS LONG, INSTEAD OF 16. APPARENTLY, IF THE BIT LENGTH
 TIMER IS RESET JUST AFTER A CLOCKING
 EDGE, THE JUST-UNDER-1-CLOCK-PULSE DELAY TILL THE NEXT PULSE,
 COMBINED WITH THE SOFTWARE DELAYS, RESULT IN THE 17 CLOCKPERIOD
 START BIT. MY D FLIP-FLOP WAS GENERATING ANOTHER 0 FOLLOWING

THE START PULSE (ALWAYS A 0) EVEN WHEN THE FIRST DATA BIT WAS A 1.

THIS CHARACTERISTIC SHOULD NOT BE THOUGHT OF AS A "BUG" IN MIKBUG. WILES & FELIX (FOOTNOTE 1) SHOW MIKBUG CONTROLLING A

FOOTNOTE 1: WILES AND FELIX, "ENGINEERING NOTE 100, MCM6830L7 MIKBUG/MINIBUG ROM", MOTOROLA SEMICONDUCTOR PRODUCTS, INC

PROGRAMMABLE COUNTER DIVIDING AN INTERNALLY GENERATED CLOCK BY 16,384 AND 32,768, RATHER THAN BY 8 AND 16, AS IN THE SWTP6800 SERIAL CONTROL INTERFACE. ON THE OTHER HAND, THE MATTER OF 17 CLOCK PULSES FOR THE START BIT IS OF NO CONSEQUENCE TO ANY WELL DESIGNED UART. ONE APPROACH TO A FIX WOULD HAVE BEEN TO INCLUDE A UART IN MY CASSETTE INTERFACE TO RECLOCK THE DATA. BUT THAT WOULD HAVE MADE IT MUCH MORE COMPLEX.

AFTER SOME THOUGHT, I CONCLUDED THAT THE ONLY OTHER WAY AROUND THIS WAS TO FORCE MIKBUG TO FIRST OUTPUT A 1, TO GET THE TIMER STARTED PROPERLY. MY MODEM WOULD IGNORE THE 1, AS IT ASSUMES THE TIME BETWEEN WORDS IS ANY RANDOM NUMBER OF 4300 HZ CLOCK PERIODS, AND THE 17 PULSE LONG 1 BIT WOULD BE INDISTINGUISHABLE FROM THE INTERWORD MARK (1) CONDITION ON THE RS232 LINE.

I WROTE A SHORT PROGRAM, BASED ON MIKBUG'S PROGRAMMING, AND MAKING MAXIMUM USE OF MIKBUG SUBROUTINES. IT IS SHOWN IN LISTING NO 1. IT CAN BE LOADED INTO RAM IN ANY CONVENIENT LOCATION. PROGRAMS FOR WHICH 1200 BAUD OUTPUT IS DESIRED MUST JUMP TO THIS SUBROUTINE RATHER THAN TO MIKBUG'S "OUTEE". I HAVE WRITTEN IT INTO MY 8 K BASIC, AND INTO THE EDITOR/ASSEMBLER I'M USING. I'VE ALSO WRITTEN AN OPERATING SYSTEM, OF SORTS, AUGMENTING MIKBUG'S FUNCTIONS, AND THIS 1200 BAUD PROGRAM IS INCLUDED THEREIN.

Editors note:

This article will be completed in the April issue of ACS I/O Port.

I wish to thank Russ Yost for this the first article in this newsletter and to try and incourage fellow members to try their luck at writting similar articles. Thanks again.

The Editor

LOC B1 B2 B3

```

357C      >*****
357C      >*
357C      >*   SERIAL OUTPUT   *
357C      >*   SUBROUTINE FOR  *
357C      >*   1200 BAUD       *
357C      >*
357C      >*   R YOST, NOV '77  *
357C      >*
357C      >*****
357C      >
E1A5      >SAV   EQU    $E1A5  MIKBUG SR. USED IN MIKBUG OUTEEE SR.
E1F3      >DE    EQU    $E1F3  MIKBUG TIMER SR.
E1EF      >DEL    EQU    $E1EF    DITTO
E1DA      >RET    EQU    $E1DA  ENTRY POINT INTO MIKBUG OUTEEE SR.
357C      >
2000      >      ORG    $2000  ARBITRARY ORIGIN. MAY BE SET ANYWHERE.
2000 37    >OUTH  PSHB      SAVE B REGISTER
2001 BD E1 A5 >      JSR    SAV    SAVE X REG. & SET IT TO PIA ADDR.
2004 C6 0A    >      LDAB   #0A    SET UP COUNTER FOR 10 TOTAL BITS
2006 BD E1 F3 >      JSR    DE    START TIMER AND DELAY FOR 1 BIT, WHILE
2009 BD E1 EF >      JSR    DEL    OUTPUTTING A "WARMUP" 1.
200C 6A 00    >      DEC    C,X    SET START BIT = 0.
200E 7E E1 DA >      JMP    RET    JUMP INTO MIKBUG OUTEEE SR. FOR REST OF
2011      >*      DATA.
2011      >      END

```

<<< UNRESOLVED ITEMS >>>

<<< SYMBOLS >>>

```

DE  E1F3  DEL  E1EF  OUTH 2000  RET  E1DA  SAV  E1A5
C

```